



Cummings J, Turska M, Rahtz S. [Challenging the Myth of Presentation in Digital Editions](#). *Journal of the Text Encoding Initiative* 2016, (9). DOI: 10.2146/ajhp150156

Copyright:

For this publication a [Creative Commons Attribution 4.0 International license](#) has been granted by the author(s) who retain full copyright.

DOI link to article:

[10.2146/ajhp150156](https://doi.org/10.2146/ajhp150156)

Date deposited:

13/09/2017



This work is licensed under a [Creative Commons Attribution 4.0 International License](#)



Journal of the Text Encoding Initiative

Issue 9 | 2016

Selected Papers from the 2014 TEI Conference

Challenging the Myth of Presentation in Digital Editions

Magdalena Turska, James Cummings and Sebastian Rahtz



Publisher
TEI Consortium

Electronic version

URL: <http://jtei.revues.org/1453>

DOI: 10.4000/jtei.1453

ISSN: 2162-5603

Electronic reference

Magdalena Turska, James Cummings, and Sebastian Rahtz, « Challenging the Myth of Presentation in Digital Editions », *Journal of the Text Encoding Initiative* [Online], Issue 9 | 2016, Online since 24 September 2016, connection on 23 May 2017. URL : <http://jtei.revues.org/1453> ; DOI : 10.4000/jtei.1453

For this publication a Creative Commons Attribution 4.0 International license has been granted by the author(s) who retain full copyright.

Challenging the Myth of Presentation in Digital Editions

Magdalena Turska, James Cummings, and Sebastian Rahtz

- 1 The number of publicly accessible digital editions is constantly growing, but only a relatively small percentage of them make their encoded source files openly available (Franzini 2016). Without the sources we cannot hope for the much-anticipated and commonly advertised re-use of all this painstakingly collected and prepared content in innovative research, visualization, and popularization.

1. “What is it Going to Look Like?”

- 2 Many (or indeed most) digital editions are created by people whose scholarly background is in textual editing. Therefore, the encoding phase is perceived only as an unavoidable step towards the real goal: the published edition, be it printed or presented otherwise.
- 3 For large digital scholarly editions, the bulk of the work is in researching and creating the underlying data, so editors sometimes think that after the encoding is complete, the rest should be trivial. At the same time, they brace themselves for the long struggle to get minute details of presentation just right. The question one hears most often during the encoding stage is: “What

is it going to look like?” And somehow the answer “Any way you like” just doesn’t seem to be understood or satisfy people. Another typical question is “How do I encode this to make it look like *this or that?*.” The latter should always ring the alarm and lead to serious discussion of editorial and encoding principles bearing in mind that honesty is one of the most important qualities of an editor.

2. Data is the Important Long-term Outcome

- 4 We would like to suggest that the encoding policy design (consisting of a schema and a set of local guidelines) and the later application of said policy to annotate a text are the most important acts that make all further research and long-term preservation of editors’ wealth of knowledge (not to mention publication) possible. Therefore in digital editions the encoded texts themselves are the most important long-term outcome of the project, while their initial presentation within a particular application should be considered only a single perspective on the data. Any given view will be far from unique or canonical, as different usage scenarios call for different presentations—ranging from “reading text” to “interactive version” with popup content, to chart, graph, or map representations and beyond. Furthermore, all initial presentations are also ephemeral, bound to be either modified over time as technologies and forms of digital publishing change, or languish in obsolescence on a forgotten server.

3. Editors will only Switch Focus to Quality of Encoding if Publication Becomes as Straightforward as Using a Text Processor

- 5 In practice the perception of value is very different. For the majority of cases custom processing of encoded documents is outside the reach of a typical editor and inevitably involves asking for technical help—which requires money and other resources, and comes with inevitable delays and communication problems. And presentation is important as that is what other scholars, funding bodies, students, and the general public will see and respond to. If editors were able to change the presentation of deeply encoded materials with a degree of self-assurance resembling their skills

with text processors, they could accept the point of view that good encoding is what ultimately counts the most. Then they might be more eager to share the encoded files, rather than just the output presentation, as the goal of their work.

- 6 Does it mean we just need better tools? A legion of editors dream about “one tool that does it all,” preferably with a nice graphic interface that hides the ugliness of raw XML and makes the frustration of dealing with formal programming languages go away.
- 7 The present situation is that there is some progress, and numerous digital humanities centers build custom workflows and in-house publishing systems, but often the use of these is limited to the host institution. Even making the infrastructure publicly available does not result in greater popularity and wider adaptation of the tools, as the case of the Kiln (formerly known as xMod) package developed at King’s College London and used practically exclusively there illustrates.¹
- 8 Meanwhile, as Tara Andrews points out:

Consensus is indeed lacking on what exactly a digital critical edition should be. As long as there is no agreement on the end result of digital philology, there can be none on its methods; as long as there is no consensus on method, there will not be widely applicable computational tools available to help produce digital critical texts.

(2013, 62)

- 9 It is highly probable that such a consensus is, for various reasons, not achievable and that therefore no simple and universal editorial environment will materialize any time soon. This is an inherent consequence of the individuality of research and the diversity of the source material that is chosen as subject matter for digital editions and virtual archives. Thus, no matter how good the infrastructure and adoption of standard vocabularies may be, it will never become the ultimate solution, as no tool can cater for all the unknown features of innovative research projects.
- 10 It seems quite telling that software development that has the biggest influence on digital humanities often takes place elsewhere and evolves with more general applications in mind: XML databases, search and indexing engines, XSLT processors, and visualization libraries; even the XML editors we use are never specifically designed to serve only the purposes of digital editions. This is not a bad thing in itself, but the natural consequence is that we need a customization layer on top of such technologies, as TEI framework does for oXygen editor, for example, to aid our particular goals.

- 11 But before we start creating such a customization, perhaps we should take another look at the general software development scene and draw lessons from there. As our lives become more and more tied to electronic devices and we grow fond of and dependent on dozens of applications we use every day, we often forget that they were most probably built within some application framework. The framework-based approach to development helps programmers to devote their time to the specifics of their project rather than dealing with the typical low-level tasks necessary for building a working application, thereby reducing overall development time. Two popular definitions illustrate important aspects of what a framework is:

A software framework is a concrete or conceptual platform where common code with generic functionality can be selectively specialized or overridden by developers or users.

(Techopedia: Software Framework)

In computer programming, a *software framework* is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software. A software framework is a universal, reusable software environment that provides particular functionality as part of a larger software platform to facilitate development of software applications, products and solutions.

(Wikipedia: Software Framework)

- 12 The key concepts here are *abstraction* and the notion of *platform offering generic functionality with room for customization*. Such an abstraction layer still requires from the editor some programming and design skills, as well as good understanding of the input data. Nevertheless, the advantages include common conventions and default behaviour that does not need to be explicitly stated and requires extending only when particular projects have different needs. This significantly reduces development time and effort, while efficient encapsulation of underlying libraries and technologies reduces the developer's learning curve and as a result leaves a much leaner and more standardized codebase to maintain.
- 13 Is there a place for a similar approach in digital editions? There seems to be no reason why there should not be.

Of course, documents worth encoding in TEI are very different from customer letters. But not that different, and eight out of ten probably will benefit from staying within the confines of a well thought-out standard schema and its surrounding processing rules. And even the two that don't may benefit from staying within that standard schema as far as possible.

(Mueller 2013)

- 14 Here Mueller hints at the idea of having a standard, re-usable processing system. TEI seems to be particularly successful as a common vocabulary, perhaps because it does not assume any ideological position about methodologies but proposes a default schema and guidelines, while always allowing customization and extension whenever projects need something that TEI does not deliver. Yet precisely because of that, the processing and publishing of TEI-encoded files is mostly left to the editor, who is typically unprepared to handle the technical aspects involved. It is usually a tough compromise between the individuality of research and the reality of the world in which computer programs do not write themselves. Could this problem be solved, perhaps, not by creating a particular piece of specialized publishing software, but rather by creating a general framework for processing TEI documents?
- 15 The most recent attempt at turning the TEI *vocabulary* into a TEI *framework* with a defined processing model has been undertaken by the TEI Simple² project. This is not the place to describe the rationale for the development of the TEI Processing Model in detail here, as the project participants are working on another article devoted solely to that subject based on the paper presented at the TEI Conference 2015 in Lyon. Suffice it to say that it creates an abstract layer for processing TEI documents which can be defined with the TEI vocabulary itself, and comes with built-in processing defaults for all TEI Simple elements. Even though the scope of TEI Simple is only a subset of the TEI vocabulary suitable for representing early-modern and modern printed material, the ideas behind the TEI Processing Model documentation lend themselves very well to the purposes of processing any TEI or even any XML document, thus making TEI Simple something very different from the earlier TEI Lite project. The Processing Model framework developed as part of the TEI Simple project hides the complexity of transforming XML documents into other formats behind higher-level interfaces through which editors can express their decisions about processing in the familiar language of TEI XML without having any knowledge of the specific target media or

processing implementation. It admittedly still requires very basic understanding of technologies like XPath and CSS, but the bar is set much lower when it comes to tweaking default processing rules, as compared with setting up a transformation system from scratch in XSLT or XQuery.

- 16 The TEI Processing Model of course is not the complete solution, for at least two reasons. First, at the current stage it is still a proposal, without the user base that can ultimately confirm its viability, even though results from early adopters like *SARIT* or *Buddhist Stonesutras* or experiments with *EEBO-TCP*³ are more than promising (see, for example, [Wicentowski and Meier 2015](#)). Second, and more important, the Processing Model covers only the document transformation aspects of an edition; building a working application on top of it still remains a significant challenge for the editorial teams, though general-purpose application frameworks, like html templating for XQuery applications on top of eXistdb, are already there to help with that process. Nevertheless, we believe that the Processing Model is a crucial step in the right direction, addressing the greatest challenge in the publication process, and that it stands a good chance of gaining more traction and becoming part of the infrastructure and recommendations maintained by the TEI Consortium. It will be practical to incorporate the Processing Model into widely used application frameworks, resulting in a promising technology stack that truly empowers editors, as the U.S. Department of State's Office of the Historian's recent adoption⁴ of a Processing Model library for eXistdb has demonstrated very clearly ([Wicentowski and Meier 2015](#)). There is no reason why this exercise could not be successfully repeated for other XML database systems such as BaseX. Thriving infrastructure projects like TAPAS⁵ and broad research networks like DiXiT⁶ would be natural targets for early adoption not only of TEI Simple, but also of the architecture and design principles it builds upon. As a result we could arrive at a flexible layered model of interlinked software packages to create a robust workflow for the creation, publication, and reuse of scholarly resources.

BIBLIOGRAPHY

- Andrews, Tara L. 2013. "The Third Way: Philology and Critical Edition in the Digital Age." *Variants* 10: 61–76. https://lirias.kuleuven.be/bitstream/123456789/352304/2/variants_postprint.pdf.
- Franzini, Greta. 2016 "A Catalogue of Digital Editions." Web. https://github.com/gfranzini/digEds_cat.

- Mueller, Martin. 2013. "TEI-Nudge or Libraries and the TEI." *Blog of the Center for Scholarly Communication & Digital Curation*, October 1. <http://sites.northwestern.edu/cscdc/?p=872>.
- "Software Framework," *Techopedia*, accessed January 16, 2016, <http://www.techopedia.com/definition/14384/software-framework>.
- "Software Framework," *Wikipedia*, accessed January 16, 2016, https://en.wikipedia.org/wiki/Software_framework.
- Wicentowski, Joseph C., and Wolfgang Meier. 2015. "Publishing TEI Documents with TEI Simple: A Case Study at the U.S. Department of State's Office of the Historian." Abstract only. In *Proceedings of Balisage: The Markup Conference 2015*. Balisage Series on Markup Technologies 15. <http://www.balisage.net/Proceedings/vol15/html/Wicentowski01/BalisageVol15-Wicentowski01.html>. doi:10.4242/BalisageVol15.Wicentowski01.

NOTES

- 1 Jose Miguel Vieira, and Jamie Norrish, "Kiln," accessed February 11, 2016, <https://github.com/kcl-ddh/kiln>.
- 2 TEI Simple, GitHub, <http://teic.github.io/TEI-Simple/>.
- 3 Early English Books Online eXist-db app, accessed February 11, 2016, <http://showcases.exist-db.org/exist/apps/eebo/works/>.
- 4 U.S. Department of State, Office of the Historian, accessed February 11, 2016, <https://history.state.gov/>.
- 5 TAPAS Project, accessed February 11, 2016, <http://www.tapasproject.org/>.
- 6 Digital Scholarly Editions Initial Training Network (DiXiT), accessed February 11, 2016, <http://dixit.uni-koeln.de/>.

ABSTRACT

Are the data of an edition means to a particular and privileged presentation, or is the presentation a side effect? Because of the changing nature of computer systems, with constant progression in hardware and software, the encoded texts are the most important long-term outcome of the project—the representation of the knowledge— and presentation within a particular application is destined to become obsolete relatively quickly.

However, it is most often the presentation output, rather than the source data, which is published and shared. We believe this is largely because there is currently no way of expressing, in the source encoding, aspects of presentation which are seen by editors as a crucial part of their work. Given a framework for encoding processing expectations for a variety of output formats, editors would be much more inclined to share the encoded files as their prime output, and intentions for presentation would be much more likely to survive repeated technology transitions as processing tools develop and change.

We believe the collision between the individuality of research and the quest for common tools that aid in the creation of digital editions will be solved not by creating another piece of specialized publishing software but rather by creating a general framework for processing TEI documents and similar, modular solutions for other tasks in the publishing workflow. Such an abstraction layer admittedly still requires some fluency in computer technologies, but far less than for setting up a publication system from scratch in a general-purpose programming language.

INDEX

Keywords: TEI, digital edition, publication framework, processing, TEI Simple

AUTHORS

MAGDALENA TURSKA

Magdalena Turska is a software developer at [eXist Solutions](#) and an elected member of the [TEI Consortium's Technical Council](#). She has recently completed her [DiXiT Marie Curie experienced researcher fellowship](#) at [IT Services](#), University of Oxford where she was a member of the TEI Simple project and one of the authors of TEI Processing Model. She was a co-editor of the [Corpus Ioannes Dantiscus' Texts and Correspondence](#). She teaches advanced TEI encoding, XSLT and XQuery and often helps projects with data modeling and application design.

JAMES CUMMINGS

James Cummings is the Senior Academic Research Technology Specialist for the [Academic IT Research Support Team](#) at [IT Services](#), University of Oxford. He is the founding director of the annual [Digital Humanities at Oxford Summer School](#) and at time of writing has been an elected member of the [TEI Consortium's Technical Council](#) since 2005. He holds a [PhD in Medieval Studies](#) from the University of Leeds and was director of [Digital Medievalist](#) from 2009 to 2012. He teaches advanced TEI encoding and customization and often helps projects with their schema design and other TEI consultation.

SEBASTIAN RAHTZ

Sebastian Rahtz had been both the Director of Academic IT (Research) and Chief Data Architect for IT Services at the University of Oxford. He was a member of the TEI Consortium's Board of Directors from 2000 to 2009, and was a member of the TEI Consortium's Technical Council for well over a decade. He was lead architect for the TEI ODD customization system in TEI P5, and wrote much of the software and infrastructure which underpins the TEI's work. He was one of the principal investigators of the TEI Simple project responsible for its overall design and execution.